requesting allocation of the resource, when the resource is already allocated to a thread

for the second process, to establish a joining function to the thread for the second process

and to provide an indication to the first process of an expected time before the resource

will become available determined based on a call duration value associated with the

second process, the joining function being operable to notify the resource manager on

termination of the thread for the second process, and the resource manager being

operable in response to termination of the thread for the second process to allocate the

resource to the thread for the first process.

17. (Twice Amended) A computer-implemented method of managing allocation of

a resource to competing processes including at least a first process and a second

process, the method including:

responding to identification of a thread for the first process requesting allocation of

the resource, when the resource is already allocated to a thread for the second process, to

establish a joining function to the thread for the second process and to provide an

indication to the first process of an expected time before the resource will become

available determined based on a call duration value associated with the second process;

responding to the joining function notifying termination of the thread for the second

process to allocate the resource to the thread for the first process.

REMARKS

In the Final Office Action dated February, 2003, the Examiner rejected claims 1,

10-11, and 17 under 35 U.S.C. 103(a) as being unpatentable over <u>Chorn</u> (U.S. Patent

No. 6,275,843) in view of <u>Abe et al.</u> (U.S. Patent No. 6,052,695), and in further view of

Simor (U.S. Patent No. 5,060,150) (Office Action [hereinafter "OA"], paragraph 2); rejected claims 2-7 and 18 under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Abe et al. and Simor, and in further view of Hoffpauir (hereinafter "Hoffpauir 1") (U.S. Stat. Inv. Reg. No. H1,896) (OA, paragraph 3); rejected claims 8-9, 12-15, and 19-21 under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Abe et al. and Simor, and in further view of Hoffpauir (hereinafter "Hoffpauir 2") (U.S. Stat. Inv. Reg. No. H1,918) (OA, paragraph 4); and rejected claim 16 under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Abe et al., Simor, and Hoffpauir 2, and in further view of Jones et al. (U.S. Patent No. 5,193,110) (OA, paragraph 5).

In view of the foregoing amendment and the remarks that follow, Applicant respectfully traverses the Examiner's rejections of the claims under 35 U.S.C. § 103(a).

The Examiner rejected claims 1, 10-11, and 17 under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Abe et al., and in further view of Simor (OA, paragraph 2). The Examiner alleged that Chorn discloses at least a resource manager operable to control the allocation of a resource to competing computing processes, the resource manager being responsive to identification of a thread for a first process requesting resource allocation. The Examiner admitted that Chorn does not disclose a joining function that notifies a resource manager on termination of a thread for a second process but alleged that the teachings of Abe et al. suggest including a joining function. The Examiner also admitted that Chorn fails to disclose "that after termination of [the] thread for a second process, the resource of the first process is allocated." The Examiner relied on Simor to teach that feature.

Claim 1, as amended herein, provides for a resource manager operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

Applicant respectfully submits that Chorn in view of Abe et al., and in further view of Simor do not disclose or suggest at least this claimed combination of elements. For example, the references do not disclose or suggest at least a resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process.

Chorn discloses a system in which multiple service requests of a global transaction are processed by a single server application program instance without the use of multiple intermediate communication server instances (abstract). A client application program provides access to one or more resources that are required by a

transaction (col. 6, line 64 - col. 7, line 1). A resource manager provides access to a resource for the application program (col. 7, lines 8-9). A communication resource manager controls communication between the client application program and other application programs that are participating in transactions (col. 7, lines 22-25). Thread identifiers are used by the communication resource manager to manage its resources for processes using the services of the communication resource manager (col. 14, lines 38-40).

In contrast, systems and methods consistent with the present invention with the present invention as recited for example in claims 1, 10-12, and 17, include a resource manager that provides an indication to a first process of an expected time before a resource will become available determined based on a call duration value associated with the second process. The call duration value may be calculated in a number of ways, including using a heuristic method based on an application type of the second process. For example, the dispatcher knows the basic type of each telephony application by virtue of the application's priority number (page 14, lines 25-27). Based on the type of application currently in control of the resource, the dispatcher can guess typical values of a call duration (page 14, lines 28-29). The resource managers of <u>Chorn</u> do not provide such an indication to a process determined based on a call duration value. Accordingly, <u>Chorn</u> does not disclose, teach, or suggest a resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an

indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process.

Abe et al. are not sufficient to overcome the aforementioned deficiencies of Chorn. Abe et al. disclose a system in which a first log processing thread outputs all log data including that for a second log processing thread and returns an output completion notification, when the second log processing thread goes into a waiting state for the output completion notification (col. 27, lines 35-45). When the notification is received, the second log processing thread terminates its own processing (col. 27, lines 45-47). Abe et al. are silent on a resource manager that provides an indication to a first process of an expected time before a resource will become available determined based on a call duration value associated with the second process. Accordingly, Abe et al., either alone or in combination with Chorn, do not disclose, teach, or suggest a resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process.

Simor is not sufficient to overcome the aforementioned deficiencies of Chorn and Abe et al. Simor discloses a system in which a process termination monitor may be requested at each resource allocation for a process (col. 17, lines 55-57). Resource managers are accordingly notified when the process terminates, and the resource release can be initiated by the resource manager itself (col. 17, lines 57-59). This configuration disclosed by Simor is different than the presently claimed invention. For

-8-

example, <u>Simor</u> does not provide an indication to a first process of an expected time before a resource will become available determined based on a call duration value associated with a second process. Accordingly, <u>Simor</u>, either alone or in combination with <u>Chorn</u> and/or <u>Abe et al.</u>, does not disclose, teach, or suggest a resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process.

For at least the foregoing reasons, Applicant submits that claim 1 is patentable over <u>Chorn</u> in view of <u>Abe et al.</u>, and in further view of <u>Simor</u>. Applicant also submits that the teachings of <u>Hoffpauir 1</u> are insufficient to remedy the aforementioned deficiencies of <u>Chorn</u>, <u>Abe et al.</u>, and <u>Simor</u> Accordingly, the combination of <u>Chorn</u>, <u>Abe et al.</u>, <u>Simor</u>, and <u>Hoffpauir 1</u> does not teach or suggest claim 1. Because claims 10-11, and 17 are independent claims with limitations similar to those of claim 1, Applicant further submits that claims 10-11, and 17 are patentable over <u>Chorn</u> in view of <u>Abe et al.</u>, and in further view of <u>Simor</u> for at least the reasons given with respect to claim 1.

The Examiner rejected claims 8-9, 12-15, and 19-21 under 35 U.S.C. 103(a) as being unpatentable over <u>Chorn</u> in view of <u>Abe et al.</u> and <u>Simor</u>, and in further view of <u>Hoffpauir 2</u>. Claim 12, as amended herein, provides for a telecommunications apparatus, comprising: at least one telephony resource for connection to a telecommunications network; and a resource manager for controlling allocation of the

telephony resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

Applicant respectfully submits that Chorn in view of Abe et al. and Simor, and in further view of Hoffpauir 2 do not disclose or suggest at least this claimed combination of elements. For example, the references do not disclose or suggest at least a resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process.

As explained above with reference to claim 1, Chorn in view of Abe et al. and Simor do not teach a resource manager being operable to provide an indication to a first process of an expected time before a resource will become available determined based on a call duration value associated with a second process. Hoffpauir 2 is not sufficient to

-10-

overcome the aforementioned deficiencies of <u>Chorn</u>, <u>Abe et al.</u>, and <u>Simor</u>. <u>Hoffpauir 2</u> disclose a system in which an integrated authentication center includes an application process, such as a call processing application, that includes a plurality of software objects such as a home location register and an authentication center (abstract). The system includes a resource manager application that manages and allocates the resources of a resource assembly with respect to a call processor and enables different applications of the call processor to interface with resources of the resource assembly (col. 13, lines 62-66).

<u>Hoffpauir 2</u> fails to show providing an indication to a first process of an expected time before a resource will become available determined based on a call duration value associated with a second process. Accordingly, <u>Hoffpauir 2</u>, either alone or in combination with <u>Chorn</u>, <u>Abe et al.</u>, and <u>Simor</u>, do not disclose, teach, or suggest a resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process.

For at least the foregoing reasons, Applicant submits that claim 12 is patentable over <u>Chorn</u> in view of <u>Abe et al.</u> and <u>Simor</u>, and in further view of <u>Hoffpauir 2</u>. Applicant also submits that the teachings of <u>Jones et al.</u> are insufficient to remedy the aforementioned deficiencies of <u>Chorn</u>, <u>Abe et al.</u>, <u>Simor</u>, and <u>Hoffpauir 2</u>. Accordingly,

the combination of <u>Chorn</u>, <u>Abe et al.</u>, <u>Simor</u>, <u>Hoffpauir 2</u>, and <u>Jones et al.</u> does not teach or suggest claim 12.

The dependent claims 2-9, 13-16, and 18-21 are allowable not only for the reasons stated above with regard to their respective allowable base claims, but also for their own patentable features that distinguish them from any combination of <u>Chorn</u>, <u>Abe et al.</u>, <u>Simor</u>, <u>Hoffpauir 1</u>, <u>Hoffpauir 2</u>, and <u>Jones et al.</u>

Since each of the claims is allowable, Applicant respectfully requests the timely allowance of this application.

Applicant respectfully requests that this Amendment under 37 C.F.R. § 1.116 be entered by the Examiner, placing claims 1-21 in condition for allowance. Applicant submits that the proposed amendments of the claims do not raise new issues or necessitate the undertaking of any additional search of the art by the Examiner. Therefore, this Amendment should allow for immediate action by the Examiner.

Furthermore, Applicant submits that the entry of the Amendment would place the application in better form for appeal, should the Examiner dispute the patentability of the pending claims.
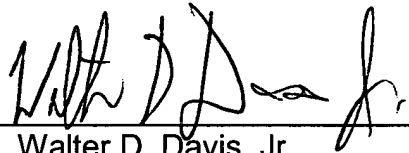
If an extension of time under 37 C.F.R. § 1.136 is required to obtain entry of this Amendment, such extension is requested. If there are any other fees due under 37 C.F.R. §§ 1.16 or 1.17 which are not enclosed herewith, including any fees required for an extension of time under 37 C.F.R. § 1.136, please charge such fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: May 12, 2003

By:_____
Walter D. Davis, Jr.
Reg. No. 45,137

-13-

APPENDIX TO AMENDMENT OF MAY 12, 2003

AMENDMENTS TO THE CLAIMS:

1. (Twice Amended) A resource manager operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

10. (Twice Amended) A resource manager operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager comprising:

means responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process; and

-14-

means responsive to the joining function notifying the resource manager on termination of the thread for the second process to allocate the resource to the thread for the first process.

11. (Twice Amended) A computer software resource manager on a data carrier, the resource manager being operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process <u>and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process</u>, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

12. (Twice Amended) <u>A telecommunications</u> [Telecommunications] apparatus<u>,</u> comprising<u>:</u>

at least one telephony resource for connection to a telecommunications network<u>;</u> and

a resource manager for controlling allocation of the telephony resource to competing computing processes including at least a first process and a second process,

the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process_ and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

17. (Twice Amended) A computer-implemented method of managing allocation of a resource to competing processes including at least a first process and a second process, the method including:

responding to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process and to provide an indication to the first process of an expected time before the resource will become available determined based on a call duration value associated with the second process;

responding to the joining function notifying termination of the thread for the second process to allocate the resource to the thread for the first process.